IN THE UNITED STATES PATENT AND TRADEMARK OFFICE

In Re Application of: Jason Alexander CU, et al.	Confirmation No. 6671
Serial No: 09/820,451	Group Art Unit: 2161
Filed: March 28, 2001	Examiner: Te Y. CHEN

For: METHOD AND SYSTEM FOR PROVIDING A GENERIC SCALAR FUNCTION

MAIL STOP APPEAL BRIEF – PATENTS
Commissioner for Patents
P.O. Box 1450
Alexandria, VA 22313-1450

SUBSTITUTE APPEAL BRIEF

Dear Sir or Madam:

In response to the Office Communication mailed February 27, 2007, Appellant submits this Substitute Appeal Brief pursuant to 37 C.F.R. § 41.37.

I. REAL PARTY IN INTEREST

The real party in interest is International Business Machines Corp. of Armonk, New York by virtue of an assignment from the inventors recorded in the U.S. Patent and Trademark Office on June 18, 2001, at Reel No. 011916, Frame No. 0070.

II. RELATED APPEALS AND INTERFERENCES

There are no appeals, interferences, or judicial proceedings known to Appellant, the Appellant' s legal representative, or Assignee, which may be related to, directly affect, be directly

affected by, or have a bearing on the decision by the Board of Patent Appeals and Interferences in the pending appeal.

III. STATUS OF CLAIMS

Claims 1-26 have been rejected. Appeal is taken from the rejection of claims 1-26.

IV. STATUS OF AMENDMENTS

An amendment after final was filed on June 19, 2006 amending claims 1-3, 6, 8-10, 13, 15-16, and 25. The amendment after final was not entered.

V. SUMMARY OF CLAIMED SUBJECT MATTER

The present invention provides a method, system and computer-readable media for utilizing a column function for a relational database in a structure query language (SQL) environment. In particular, the method, system and computer-readable media allow conventional, pre-existing column functions to operate on data other than columns.

Relational databases organize data into entries that are placed into one or more columns and one or more rows of a table. Such a relational database is typically used to archive information. Specification, page 1, lines 7-14. The column function is a conventional column function that is already in existence and is used in performing an operation on an indeterminate number of entries in a table of a relational database. More specification, a column function is a pre-existing function that typically operates on data that has been organized into one or more columns. Specification, page 3, lines 1-3. The existence of and operation of the pre-existing

column function is already known. Specification, page 2, lines 11-17 and page 3, lines 8-12 (giving as examples of column functions the minimum, maximum, sum and average functions).

In the method, system, and computer-readable medium in accordance with the present invention, a generalized scalar function is used to allow other row data to be operated on by a column function. At least one row of data is specified as an argument for the generalized scalar function. Specification, page 7, lines 21-23. The generalized scalar function is used to simulate a column environment for the row(s) that is the argument for the generalized scalar function. Specification, page 8, lines 10-13. For example, Figure 4 describes one embodiment of a method that can be used for data in a table such at the table 1 depicted in Figure 1. In such an embodiment, a user specifies which row(s) is to be used for data. Specification, page 9, lines 12-13 and Figure 4, item 152. Stated differently, the row(s) are specified as the argument for the generalized scalar function. Referring to Figures 1 and 4, suppose that the user specifies that the row 6 of the table 1 is the row to be operated on. The generalized scalar function fetches the first entry (11) and provides the entry to the column function. Specification, page 9, lines 13-18 and Figure 4, items 154 and 156. The column function then operates on the entry normally. Specification, page 9, line 17-page 10, line 1 and Figure 4, items 158, 160, 166 and 168. It is determined whether the entry provided to the column function is the last in the row and, if not, the generalized scalar function provides the next entry (i.e. entry 11) to the column function. Specification, page 10, lines 1-4, Figure 4, item 164. This process repeats until the last entry (e.g. entry 14) is provided to the column function. Once the last entry is provided, then the column function completes its (normal and conventional) operations and provides the output. Specification, page 10, lines 4-6 and Figure 4, items 166 and 168. If other rows are also to be

processed, then a similar method is carried out for additional rows. Specification, page 10, lines 6-8. Thus, the data in the row is provided to the column function entry by entry, as though the row was a column.

Similarly, as indicated in Paragraph 7 of the Declaration under 37 C.F.R. 1.132 (Declaration), the entries in the row could simply be provided one-by-one to the column function. See also, specification, page 9, lines 15-18 and page 10, lines 1-3. Thus, the generalized scalar function allows data in the row(s) to be provided to the column function as though the row(s) were column(s). Specification, page 8, lines 15-17. The column function is performed on the data in the row(s) to provide the output(s) of the column function. Stated differently, the column function operates in a conventional manner on the row entries provided to the column function. Specification, page 8, lines 17-20. Thus, the generalized scalar function is used to provide the entries in a row to the column function as though the entries came from a column. Specification, page 8, lines 15-17. As a result, in addition to performing operations in a conventional manner on column data, the column function can perform its operations on row(s). Specification, page 8, lines 17-19. Consequently, column functions, such minimum, maximum, sum, and average can be performed on rows without requiring the column functions to be rewritten. Specification, page 8, lines 1-6. As a result, resources are conserved.

Thus, the methods (depicted in Figures 3-4), systems (depicted in Figure 5), and computer-readable media recited in claims 1-21 may provide row data entry by entry to the column function. Stated differently, the generalized scalar function simulates the column environment such that the data in the row can be provided to the column function as though the row were a column. The pre-existing and conventional column function operates normally.

Specification, page 8, lines 17-19. As a result, the operations of conventional column function can be performed on row data. The conventional column function can, therefore, be used on rows without rewriting the conventional column function. Specification, page 9, lines 1-4. The resources that would otherwise be expended on rewriting and testing the column function can be saved. Specification, page 8, lines 4-6.

Independent claim 1 recites a method for utilizing a column function (202) for a relational database in a structure query language (SQL) environment, the column function (202) capable of performing an operation on an indeterminate number of entries, the relational database utilizing data including a plurality of entries being organized into at least one column (2,3,4,5) and at least one row (6,7,8). *See, e.g.*, pg. 6, ln. 17 to pg. 7, ln. 5; pg. 7, ln. 21 to pg. 9, ln. 6; pg. 10, ln. 14 to pg. 11, ln. 3; FIGs. 1, 3, and 5. The method includes allowing a user to specify the at least one row (6,7,8) as an argument for a generalized scalar function (204) (102). *See, e.g.*, pg. 6, lns. 21-22; pg. 7, lns. 21-22; pg. 10, lns. 17-20; FIGs. 1, 3, and 5. The method also includes simulating a column environment for the at least one row (6,7,8) using the generalized scalar function (204) to allow the at least one row (6,7,8) to be provided to the column function (202) as though the at least one row (6,7,8) was a column (104). *See, e.g.*, pg. 6, ln. 23 to pg. 7, ln. 2; pg. 8, lns. 10-11; pg. 10, lns. 20-22; FIGs. 1, 3, and 5. The method further includes performing the column function (202) on the at least one row (6,7,8) to provide at least one output (106). *See, e.g.*, pg. 7, lns. 2-3; pg. 8, lns. 17-19; pg. 10, ln. 23 to pg. 11, ln. 3; FIGs. 1, 3, and 5.

Independent claim 8 recites a computer-readable medium containing a program for utilizing a column function (202) for a relational database in a structure query language (SQL) environment, the column function (202) capable of performing an operation on an indeterminate

number of entries, the relational database utilizing data including a plurality of entries being organized into at least one column (2,3,4,5) and at least one row (6,7,8). *See, e.g.*, pg. 6, ln. 17 to pg. 7, ln. 5; pg. 7, ln. 21 to pg. 9, ln. 6; pg. 10, ln. 14 to pg. 11, ln. 3; FIGs. 1, 3, and 5. The program includes instructions for allowing a user to specify the at least one row (6,7,8) as an argument for a generalized scalar function (204) (102). *See, e.g.*, pg. 6, lns. 21-22; pg. 7, lns. 21-22; pg. 10, lns. 17-20; FIGs. 1, 3, and 5. The program also includes instructions for simulating a column environment for the at least one row (6,7,8) using the generalized scalar function (204) to allow the at least one row (6,7,8) to be provided to the column function (202) as though the at least one row (6,7,8) was a column (104). *See, e.g.*, pg. 6, ln. 23 to pg. 7, ln. 2; pg. 8, lns. 10-11; pg. 10, lns. 20-22; FIGs. 1, 3, and 5. The program further includes instructions for performing the column function (202) on the at least one row (6,7,8) to provide at least one output (106). *See, e.g.*, pg. 7, lns. 2-3; pg. 8, lns. 17-19; pg. 10, ln. 23 to pg. 11, ln. 3; FIGs. 1, 3, and 5.

Independent claim 15 recites a system for utilizing a column function (202) for a relational database in a structure query language (SQL) environment, the relational database utilizing data including a plurality of entries being organized into at least one column (2,3,4,5) and at least one row (6,7,8). *See, e.g.*, pg. 6, ln. 17 to pg. 7, ln. 5; pg. 7, ln. 21 to pg. 9, ln. 6; pg. 10, ln. 14 to pg. 11, ln. 3; FIGs. 1, 3, and 5. The system includes a column function (202) capable of performing an operation on an indeterminate number of entries. *See, e.g.*, pg. 6, lns. 18-19; FIG. 5. The system also includes a generalized scalar function (204) for simulating a column environment for the at least one row (6,7,8) using the generalized scalar function (204) to allow the at least one row (6,7,8) to be provided to the column function (202) as though the at

least one row (6,7,8) was a column such that the column function (202) can perform an operation the at least one row (6,7,8) to provide at least one output. *See, e.g.*, pg. 6, ln. 23 to pg. 7, ln. 3; pg. 8, lns. 10-11 and 17-19; pg. 10, lns. 20 to pg. 11, ln. 3; FIGs. 1, 3, and 5. The system further includes an interface (206) for allowing a user to specify the at least one row (6,7,8) as an argument for the generalized scalar function (204). *See, e.g.*, pg. 6, lns. 21-22; pg. 7, lns. 21-22; pg. 10, lns. 17-20; FIGs. 1, 3, and 5.

Independent claim 25 recites a method for utilizing a column function (202) for a relational database in a structure query language (SQL) environment, the column function (202) capable of performing an operation on an indeterminate number of entries, the relational database utilizing data including a plurality of entries being organized into at least one column (2,3,4,5) and at least one row (6,7,8), each of the at least one row (6,7,8) including a plurality of entries. *See, e.g.*, pg. 6, ln. 17 to pg. 7, ln. 5; pg. 7, ln. 21 to pg. 9, ln. 6; pg. 10, ln. 14 to pg. 11, ln. 3; FIGs. 1, 3, and 5. The method includes allowing a user to specify the at least one row (6,7,8) as an argument for a generalized scalar function (204) (102). *See, e.g.*, pg. 6, lns. 21-22; pg. 7, lns. 21-22; pg. 10, lns. 17-20; FIGs. 1, 3, and 5. The method also includes simulating a column environment for the at least one row (6,7,8) using the generalized scalar function (204) to allow the at least one row (6,7,8) to be provided to the column function (202) as though the at least one row (6,7,8) was a column, the simulating further including using the generalized scalar function (204) to provide the plurality of entries for each of the at least one row (6,7,8) to the column function (202) entry by entry (104,156). *See, e.g.*, pg. 6, ln. 23 to pg. 7, ln. 2; pg. 8, lns. 10-11; pg. 9, lns. 14 to pg. 10, ln. 8; pg. 10, lns. 20-22; FIGs. 1, 3-5. The method further includes performing the column function (202) on the at least one row (6,7,8) to provide at least one

output (106). *See, e.g.*, pg. 7, lns. 2-3; pg. 8, lns. 17-19; pg. 10, ln. 23 to pg. 11, ln. 3; FIGs. 1, 3, and 5.

VI. GROUNDS OF REJECTION TO BE REVIEWED ON APPEAL

1. Appellant requests review as to the objections to the Specification under 37 CFR § 1.71.

2. Appellant requests review as to claims 1-26, and their rejection under 35 U.S.C. § 101 as being directed to non-statutory subject matter.

3. Appellant requests review as to claims 1-26, and their rejection under 35 U.S.C. § 112, second paragraph, as being indefinite for failing to particularly point out and distinctly claim the subject matter which applicant regards as the invention.

4. Appellant requests review as to claims 1-26, and their rejection under 35 U.S.C. § 102(e), as being anticipated by U.S. Patent No. 6,289,336 to Melton et al. (hereinafter "Melton").

VII. ARGUMENTS

A. Summary of the Applied Rejections

In the above-identified Office Action, the Examiner objected to the specification under 37 C.F.R. 1.71 as failing to provide an adequate description of the invention. In so doing, the Examiner stated that the specification fails to disclose "the actual, practical steps for forming the claimed abstract generalized scalar function, activating the claimed abstract generalized scalar function[,] initializing the first entry, evaluating each entry and finalizing the lat entry of the at least

one raw [sic], such that the simulating of a column environment will produce a useful, concrete, and tangible result."

The Examiner rejected claims 1-26 under 35 U.S.C. § 112, first paragraph, as containing subject matter which was not disclosed in the specification in such a manner as to allow one of ordinary skill in the art to make and/or use the invention. In so doing, the Examiner stated that the "'allowing' phrase repeatedly recited in independent claims 1, 8, and 15 does not cause any functionality to occur in the computer system. This is demonstrated by the absent recitation of any code or steps for causing a computer to do anything." The Examiner also indicated that in an interview, the "application attorney. . . on record indicated that the instant invention deals with reversing a matrix (or a table)."

The Examiner also rejected claims 1-26 under 35 U.S.C. § 112, second paragraph, as being indefinite for failing to point out and distinctly claim the subject matter that the Appellant regards as the invention. In so doing, the Examiner indicated that the structure and function of the generalized scalar function was indefinite.

Further, the Examiner rejected claims 1-26 under 35 U.S.C. § 102 as being anticipated by Melton.

Appellant respectfully requests that the Board reverse the Examiner's objection to the specification and the Examiner's final rejection of claims 1-26 under 35 U.S.C. § 101, 35 U.S.C. § 112 second paragraph and under 35 U.S.C. § 102.

B. The Cited Prior Art

Appellant agrees that Melton describes specific functions and a corresponding database query complier and its associated compilation methods. Melton, Abstract and Fig. 1. The cited portions of Melton describe a specific set of row functions that are written. In particular, cited portions of Melton describe row functions that are written to search previously accessed rows or offsets. Melton, col. 2, lines 10-45. As such, the functions in Melton are written to perform specific operations on rows. For example, the Rows Since function parses a search condition, identifies information from a previous row, and converts the information to provide an offset function based on the previously accessed row. Melton, col. 2, lines 10-19. The table is searched, looping through previous rows until the desire row, that satisfies the search condition, is found. Melton, col. 2, lines 20-23. The information in the row that satisfied the search condition is made available. Thus, information from a previous row can be accessed without having to change a cursor' s position in a row. Melton, col. 2, lines 42-45. Finally, Appellant also notes that in an Office Action dated March 30, 2005, the Examiner had indicated that Melton "fails to disclose a specific instance of a generalized scalar function linked to a column function (or running and moving sequence function) as specified by a user."

C. The Specification is Adequate Under 37 C.F.R. § 1.17

Appellant respectfully submits that the Examiner' s objection to the specification under 36 C.F.R. 1.171 is without merit as the Examiner has completely failed to explain why the specification is insufficient to ensure that the inventor had possession, as of the filing date of the application, the specific subject matter claimed. In particular, The Examiner' s objections appear to rest on the use of the terms "generalized scalar function" and "column function." Appellant

respectfully submits that the recited generalized scalar function, the column function that initializes, evaluates, and finalizes the resultant, and the cooperation between the generalized scalar function and the column function are described in the specification with such particularity as to ensure that the Inventor had possession of the subject matter claimed.

As is clearly indicated in the specification, a column function is a preexisting, conventional function in relational databases. Specification, page 2, line 14-page 3, 12 and page 10, lines 9-13. See also, Paragraph 4, Declaration by Jason Cu (submitted with Amendment filed September 30, 2003). Such conventional column functions operate on a column of data and return a particular result. Specification, page 2, lines 11-13. For example, one example of a column function is a minimum function, which would return the minimum value in a particular column. Specification, page 2, lines 14-17. Other examples of conventional column functions given in the specification include maximum, sum and average, which return the maximum value, the sum of the values, and the average of the values, respectively, in the column(s) on which the conventional column functions operate. Specification, page 3, lines 8-12.

In addition to providing well-known examples of column functions, in the Background of the Invention, the activities performed by the conventional column function are described. In particular, the initialization phase, evaluation phase, and finalization phase of a conventional column function are also described. Specification, page 3, lines 13-19. Thus, given the fact that column functions are conventional preexisting and well known functions, the examples of conventional column functions provided in the specification, and the description of the phases of a conventional column function, Appellant respectfully submits that one of ordinary skill in the art would readily understand that the term column function. As such, Appellant respectfully

submits that one of ordinary skill in the art would readily recognize how such conventional column functions operate, including use of the initialization, evaluation, and finalization phases.

Moreover, nothing new or different is done to the column function. See Declaration, paragraph 4. Instead, the embodiments of the method and system described in the specification illustrate a new use for a conventional column function: performing functions on row data. In order for the column function to perform this new task, the generalized scalar function operates on the row data. The manner in which the generalized scalar function operates on the row data to provide the data to the column function is described below. Consequently, the column function is a conventional function used in a conventional way. Appellant respectfully submits, therefore, that based on a reading of the specification, one of ordinary skill in the art would understand the term column function, would understand that the inventors have possession of the column function of the claimed invention, and would be capable of using the (pre-existing) column function in the manner recited in the claims and described in the specification.

The generalized scalar function is adequately described in the specification and depicted in the drawings. Moreover, the operation of the generalized scalar function in conjunction with the column function is adequately described. The generalized scalar function is used to simulate a column environment for row(s) of data so that the row can be provided to the pre-existing column function as if the row were a column. Stated differently, the generalized scalar function simulates the column environment for the row(s) so that the column function can operate on the row data. Specification, page 8, lines 14-19.

The specification describes a specific embodiment of a generalized scalar function in which data from a row is provided entry-by-entry to the column function. Specification, page 9,

line 14-page 10, line 8 and Figure 4. See also Paragraph 7 of the Declaration. As expressly stated in the specification, the generalized scalar function fetches a row that has been specified as an argument for the generalized scalar function. Specification, page 9, lines 12-15. The generalized scalar function provides a single entry from the row that has been fetched to a column function, which operates on the entry. Specification, page 9, lines 15-22. See also Specification, page 8, line 10-page 9, line 2. It is determined whether there are additional entries in the row and, if so, these are also provided entry-by-entry to the column function. Specification, page 10, lines 1-8. Moreover, the specification describes entries as corresponding to intersections of rows and columns (e.g. entry 11 in Figure 1), rather than entire rows or other entities. Specification, page 1, lines 7-14. Moreover, these operations performed by the generalized scalar function are *depicted* in Figure 4, items 152, 154, 156, 162, 164, and 170. More specifically, the specification states:

> Referring back to Figures 1 and 4, a row 6, 7 or 8 of the table 1 is fetched, via step 154. **A first entry of the row 6, 7 or 8 is provided to the conventional column function, via step 156.** . . An initialization phase for the conventional column function is carried out, via step 158. Once the initialization phase is performed or if it is determined that the entry provided is not the first entry, then an evaluation phase is performed, via step 160. Thus, the operations necessary for the conventional column function to provide an output are performed in step 160. Step 160 might include adding the data in the entry to a running sum or determining whether the data in the entry is the minimum or maximum encountered. It is determined whether the entry is the last entry in the row 6, 7 or 8, via step 162. **If [the entry just provided is] not [the last entry], the next entry in the row is provided to the conventional column function, via step 164.** The method 150 then returns to step 160 to the evaluation phase for subsequent entries. If the entry is the last in the row 6, 7 or 8, then the conventional column function enters it finalization phase, via step 166 and returns an output, via step 168. It is then determined whether there are any additional arguments in the generalized scalar function, via step 170. If so, step 152 is returned to so that the next row can be fetched. Otherwise, the method 150 terminates.

Specification, page 9, line 14-page 10, line 8 (emphasis added). Thus, providing individual entries to the column function (steps 156 and 164) and using the column function (steps 158, 160, 166, and

168) to perform functions on the entries are described. Consequently, Appellant respectfully submits that the discussion previously added regarding providing the row entry-by-entry to the column function did have support in the specification.

Because the column function receives the data entry by entry, the column function can operate on the data from the row. Specification, page 9, lines 1-4. Stated differently, the way in which the generalized scalar function provides data to the column function allows the column function to treat the entries in the row as though they are entries in a column. Figure 4, items 158, 160, 166, and 168. The column function carries out its operations in a conventional manner by performing the appropriate ones of the initialization, evaluation, and finalization phases. Specification, page 9, line 18-page 10, line 6 and Figure 4, items 158, 160, and 166. Consequently, the specification not only describes with specificity an embodiment of the generalized scalar function, but also describes and depicts the operation of this embodiment of the generalized scalar function in conjunction with the conventional column function. Accordingly, Appellant respectfully submits that one of ordinary skill in the art would understand the term generalized scalar function and how such a generalized scalar function interfaces with a conventional column function. Appellant thus respectfully submits that one of ordinary skill in the art would understand the term generalized scalar function, would understand that the inventors have possession of the generalized scalar function of the claimed invention, and would be capable of using the generalized scalar function in the manner recited in the claims and described in the specification.

For the above-identified reasons Appellant respectfully requests that the Board reverse the Examiner' s objection to the specification under 37 C.F.R. 1.171.

D. Claims 1-26 Are Not Unpatentable Under 35 U.S.C. § 101

Appellant respectfully submits that the applied rejections of claims 1-26 under 35 U.S.C. § 101 are without merit as the Examiner has completely failed to explain why the claims contain nonstatutory subject matter.

Independent claim 1 recites a method for using a "column function for a relational database in a structure query language (SQL) environment." Claim 1 also recites the use of a generalized scalar function, including the steps of allowing one or more rows of a table to specified as the argument for the generalized scalar function and using the generalized scalar function to simulate a column environment to "allow the at least one row to be provided to the column function as though the at least one row was a column." Claim 1 also recites that the column function operates on the row(s) specified as arguments for the generalized scalar function and provides an output. Independent claims 8, 15, and 25 recite analogous computer-readable medium, system, and method claims, respectively. In the rejection in the above-identified Final Office Action dated April 17, 2006, the Examiner appeared to focus on the "allowing" terms as indicating that the claims were directed to nonstatutory subject matter. In addition, the Examiner indicated that the "application attorney. . . on record indicated that the instant invention deals with reversing a matrix (or a table)." Consequently, these will primarily be discussed.

Appellant strongly disagrees with the Examiner' s contention that the "application attorney" indicated that the instant invention deals with reversing a matrix or table. In the interview cited by the Examiner, it became apparent that the Examiner was having difficulty understanding functioning of the present invention. Thus, in order to facilitate the Examiner' s understanding of the present invention, Appellant' s attorney stated that the present invention could be thought of as

transposing the matrix in that rows could be operated on by a function that normally operates on columns. In other words, the present invention allows rows to be used (by a column function) as though the rows were columns. Application attorney further stated during the interview and in remarks subsequently filed that this analogy was made in order to aid the Examiner in understanding the present invention, not to characterize the operation of the present invention. As the application attorney has repeatedly stated, the present invention recited in varying scope in claims 1-26 did not function to transpose a matrix. Moreover, Appellant respectfully submits that even if the present invention did include transposing a matrix or table in a database system, the present invention would provide a function in a computer system. This function would be to operate to change form of stored data and provide an output based on the data.

Appellant disagrees with the Examiner' s indication that the inclusion of "allowing" phrases means that the claims do not cause any functionality in a computer system. Claim 1 recites "allowing a user to specify the at least one row as an argument for a generalized scalar function." Thus, the user may provide row(s) to the generalized scalar function. However, this is not the only step recited in claim 1. Claim 1 also recites the step of "simulating a column environment for the at least one row using to generalized scalar function to allow the at least one row to be provided to the column function as though the at least one row was a column . . ." In this element, the phrase starting with "to allow" describes what simulation of the column environment achieves. Based on the embodiment described in the specification and discussed above, this "simulating step" in claim 1 means at least providing data entry-by-entry to the column function.

In addition, claim 1 also recites the step of "performing the column function on the at least one row to provide at least one output." Consequently, as recited in claim 1, an output is

provided. This output is provided from the column function and is based on the row data provided to the column function using the generalized scalar function. Thus, although claim 1 does include "allowing" phrases, claim 1 also recites steps that cause functionality including but not limited to providing an output and providing data to the column function in a specific manner. Consequently, claim 1 is allowable under 35 U.S.C. § 101.

Independent claims 8, 15, and 25 recite analogous claims. For example, claim 8 recites a computer-readable medium including analogous simulating and performing instructions and recites that at least one output is provided. Claim 15 recites an analogous system that includes a column function, a generalized scalar function and an interface. Claim 25 recites simulating and performing steps and recites that at least one output is provided. As such, the claimed invention is directed to statutory subject matter. Consequently, Appellant respectfully submits that the Examiner' s rejection under 35 U.S.C. § 101 is incorrect.

Claims 2-7 and 22, claims 9-14 and 23, claims 16-21 and 24 and claim 26 depend upon independent claims 1, 8, 15, and 25, respectively. Consequently, the arguments herein apply with full force to claims 2-7, 9-14, 16-24, and 26. Appellant thus respectfully submits that one of ordinary skill in the art would be capable of making and/or using the subject matter recited in claims 1-26 Accordingly, Appellant respectfully requests that the Board reverse the Examiner' s rejection of claims 1-26 under 35 U.S.C. § 101.

E. Claims 1-26 Are Not Unpatentable Under 35 U.S.C. § 112, Second Paragraph

Appellant respectfully submits that the applied rejections of claims 1-26 under 35 U.S.C. § 112, second paragraph, are without merit as the Examiner has completely failed to explain why

claims 1-26 are indefinite for failing to point out and distinctly claim the subject matter the Appellant regards as the invention.

Independent claims 1, 8, 15, and 25 recite the use of a generalized scalar function to simulate a column environment and a column function. When read in light of the specification, the generalized scalar function, the simulation of the column environment, and the column function are clear and definite.

As described in the specification, the column function is a conventional, column function such as the minimum, maximum, average, or sum functions. Specification, page 2, lines 11-17 and page 3, lines 8-12. Further, the column function operates in a normal, conventional manner. Specification, page 9, line 17-page 10, line 1 and Figure 4, items 158, 160, 166 and 168. Thus, the column function recited in claims 1, 8, and 15, and described in the specification is a conventional column function. Thus, recitation of the column function in claims 1, 8, and 15 would be clear and definite to one of ordinary skill in the art.

As discussed above, the generalized scalar function has a function that is recited and can be described at a high level as simulating the column environment such that the column function can operate on the entries of the row provided as an argument to the generalized scalar function. Specification, page 7, lines 21-23 and page 8, lines 10-13. The generalized scalar function is described in the specification. See, for example, Specification, page 8, line 10-page 9, line 2. See also, page 9, line 12-page 10, line 8. Thus, the form of one embodiment of the generalized scalar function is essentially depicted at a flow chart level in a portion of Figure 4. In one embodiment, the generalized scalar function performs this simulation by fetching the row and providing the entries of the row entry-by-entry to the column function. Specification, page 9, lines 15-18 and

page 10, lines 1-3. See also, Declaration, paragraph 7. Because the row data are provided entry-by-entry to the column function, the column function can operate on the row data as though the row(s) were column(s). Specification, page 8, lines 15-17 and Figures 4-5. Thus, at least one embodiment of the generalized scalar function is quite simple in nature–simply fetching a row and providing entries in the row entry-by-entry to the column function.

In addition, the cooperation of this embodiment of the generalized scalar function with the column function is described and depicted. Specification, page 9, lines 12-18; page 10, lines 1-8 and Figure 4. The specification even states that in one embodiment: "the steps 154 [fetch row], 156 [provide first entry in row to column function], and 164 [provide next entry in row to column function] . . . **are used to simulate the column environment** for the rows 6, 7, or 8 that is input as an argument to the generalized scalar function." Specification, page 9, lines 16-18. Thus, as described in the portions of the specification cited above, for row 6 of the table 1 depicted in Figure 1, one embodiment of the generalized scalar function would individually provide entries 11, 12, 13, and 14 to the appropriate column function. The column function would then operate in a conventional manner on these entries.

Thus, when read in light of the specification, the terms of claims 1, 8, 15, and 25 are clear and definite. Accordingly, Appellant respectfully requests that the Board reverse the Examiner' s rejection of claims 1, 8, 15, and 25 under 35 U.S.C. § 112, second paragraph.

Claims 2-7 and 22, claims 9-14 and 23, claims 16-21 and 24 and claim 26 depend upon independent claims 1, 8, 15, and 25, respectively. Consequently, the arguments herein apply with full force to claims 2-7, 9-14, 16-24, and 26. Further, with respect to claims 6, 13, and 20, the column function may carry out its operations in a conventional manner by performing the

appropriate ones of the initialization, evaluation, and finalization phases. Specification, page 9, line 18-page 10, line 6 and Figure 4, items 158, 160, and 166. Thus, Appellant respectfully submits that these phases are well understood by those of ordinary skill in the art. See also, Declaration by Jason Cu (submitted with Amendment filed September 30, 2003).

Appellant thus respectfully submits that claims 1-26 are clear and definite. Accordingly, Appellant respectfully requests that the Board reverse the Examiner' s rejection of claims 1-26 under 35 U.S.C. § 112, second paragraph.

F. Claims 1-26 Are Not Unpatentable Under 35 U.S.C. § 102

Appellant respectfully submits that the applied rejections of claims 1-26 under 35 U.S.C. § 102 are without merit as the Examiner has completely failed to explain why Melton teaches or suggests the methods recited in claims 1-26.

As a preliminary manner, Appellant notes that in an Office Action dated March 30, 2005, the Examiner expressly stated that Melton "fails to disclose a specific instance of a generalized scalar function linked to a column function (or running and moving sequence function) as specified by a user." Consequently, the Examiner had previously cited another reference in order to remedy the defects of Melton. Thus, despite the Examiner' s review of the record, Appellant respectfully submits that as the Examiner has previously recognized, Melton fails to teach or suggest the method, system, and computer-readable medium recited in claims 1-26.

Claim 1 recites a method including the steps of allowing a user to specify the at least one row as an argument for a generalized scalar function,

> simulating a column environment for the at least one row using the generalized scalar function to allow the at least one row to be provided to the column function

> as though the at least one row was a column; and . . . performing the column function on the at least one row to provide at least one output.

Thus, claim 1 recites that the generalized scalar function is used "to allow the at least one row to be provided to the column function as though the at least one row was a column." The column function can thus performed for data in the at least one row, which now mimics a column. Consequently, at least one output for the row data is provided by the column function. Claims 8, 15, and 25 recite analogous computer-readable medium, system, and method claims.

Thus, using the method, computer-readable medium and system recited in claims 1, 8, 15, and 25 respectively, the pre-existing column function can be reused to work on row data. As a result, the resources that would be used in rewriting, testing, and debugging a row function that performs the operations of the column function are saved. Specification, page 9, lines 1-6; page 10, lines 9-13. Consequently, the methods, computer-readable medium, and system recited in claims 1, 8, 15, and 25, can be used to extend the ability of the pre-existing column function to operate on row data without expending significant additional resources.

In contrast, the cited portions Melton fails to teach of suggest the use of the recited generalized scalar function in conjunction with a (pre-existing, conventional) column function. Instead, the cited portions of Melton describe a specific set of row functions that are written. Melton expressly states that the sequence functions described in Melton "are functions that operate on ordered sets of rows and that require knowledge of or access to past values." Melton, col. 1, lines 13-16. Thus, Melton describes row functions that are written to search previously accessed rows or offsets. Melton, col. 2, lines 10-45. The functions described in the cited portion of Melton are for rows, not columns, and to perform specific operations on these rows. Appellant can find no indication in the cited portions of Melton that the functions of Melton are

used in conjunction with pre-existing column functions. The cited portions of Melton are also devoid of mention of utilizing a generalized scalar function to simulate a column environment so that the row data appears to the column function as a column. For example, the cited portion of Melton also does not describe fetching a row and providing the row data to the column function as if the row was a client. Thus, the cited portions of Melton fails to teach or suggest using a generalized scalar function to allow the row(s) to be provided to the column function as though the at least one row was a column in conjunction with a column function that performs its operation in a conventional manner. Consequently, the cited portions of Melton fail to teach or suggest the methods, computer-readable medium and system recited in claims 1, 8, 15, and 25. Accordingly, Appellant respectfully submits that claims 1, 8, 15, and 25 are allowable over the cited references and respectfully requests that the Board reverse the final rejection of claims 1, 8, 15, and 25 under 35 U.S.C. § 102.

Claims 2-7 and 22, claims 9-14 and 23, claims 16-21 and 24 and claim 26 depend upon independent claims 1, 8, 15, and 25, respectively. Consequently, the arguments herein apply with full force to claims 2-7, 9-14, 16-24, and 26. Accordingly Appellant respectfully requests that the Board reverse the final rejection of claims 1-26 under 35 U.S.C. § 102.

G. Summary of Arguments

For all the foregoing reasons, it is respectfully submitted that the specification provides an adequate written description of the invention under 37 C.F.R. 1.71, that claims 1-26 (all the claims presently in the application) are patentable under 35 U.S.C. § 101, that claims 1-26 are patentable under 35 U.S.C. § 112 first and second paragraphs, and that claims 1-26 define subject matter which would not have been obvious under 35 U.S.C. § 103 or anticipated under 35 U.S.C.

§ 102 at the time the subject matter was invented. Thus, Appellant respectfully requests that the Board reverse the rejection of all the appealed claims and find each of these claims allowable.

Respectfully submitted,

SAWYER LAW GROUP LLP

Dated: March 13, 2007

/JOSEPH A. SAWYER, JR./

Joseph A. Sawyer, Jr.
Attorney for Applicants
Reg. No. 30,801
(650) 475-1435

APPENDIX OF CLAIMS

1. (Previously Presented) A method for utilizing a column function for a relational database in a structure query language (SQL) environment, the column function capable of performing an operation on an indeterminate number of entries, the relational database utilizing data including a plurality of entries being organized into at least one column and at least one row, the method comprising the steps of:

(a) allowing a user to specify the at least one row as an argument for a generalized scalar function;

(b) simulating a column environment for the at least one row using the generalized scalar function to allow the at least one row to be provided to the column function as though the at least one row was a column; and

(c) performing the column function on the at least one row to provide at least one output.

2. (Original) The method of claim 1, wherein the simulating step (b) further includes the steps of:

(b1) fetching a row of the at least one row; and

(b2) utilizing the generalized scalar function to provide the row to the column function as though the row was a column.

3. (Original) The method of claim 1, wherein the column function performing step (c) further includes the step of:

(c1) performing the column function on the row to provide an output; and wherein the method further includes the step of

(d) repeating steps (b1), (b2) and (c1) for each remaining row of the at least one row.

4. (Original) The method of claim 1, wherein the column function provides a maximum of each of the at least one row.

5. (Original) The method of claim 1, wherein the column function provides a minimum of each of the at least one row.

6. (Original) The method of claim 1, wherein the column function performing step (c) further includes the step of:

(c1) performing an initialization phase in response to a first entry of each of the at least one row;

(c2) performing an evaluation phase on each entry of the at least one row; and

(c3) performing a finalization phase after evaluation of a last entry of the at least one row.

7. (Previously Presented) The method of claim 1, wherein the generalized scalar function in combination with the column function allow the operation of the column function to be performed for the indeterminate number of entries in the at least one row.

8. (Previously Presented) A computer-readable medium containing a program for utilizing a column function for a relational database in a structure query language (SQL) environment, the column function capable of performing an operation on an indeterminate number of entries, the relational database utilizing data including a plurality of entries being organized into at least one column and at least one row, the program including instructions for:

(a) allowing a user to specify the at least one row as an argument for a generalized scalar function;

(b) simulating a column environment for the at least one row using the generalized scalar function to allow the at least one row to be provided to the column function as though the at least one row was a column; and

(c) performing the column function on the at least one row to provide at least one output.

9. (Original) The computer-readable medium of claim 8, wherein the simulating instructions (b) further includes instructions for:

(b1) fetching a row of the at least one row; and

(b2) utilizing the generalized scalar function to provide the row to the column function as though the row was a column.

10. (Original) The computer-readable medium of claim 8, wherein the column function performing instructions (c) further includes instructions for:

(c1) performing the column function on the row to provide an output; and wherein the program further includes instructions for

(d) repeating instructions (b1), (b2) and (c1) for each remaining row of the at least one row.

11. (Original) The computer-readable medium of claim 8, wherein the column function provides a maximum of each of the at least one row.

12. (Original) The computer-readable medium of claim 8, wherein the column function provides a minimum of each of the at least one row.

13. (Original) The computer-readable medium of claim 8, wherein the column function performing instruction (c) further includes instructions for:

(c1) performing an initialization phase in response to a first entry of each of the at least one row;

(c2) performing an evaluation phase on each entry of the at least one row; and

(c3) performing a finalization phase after evaluation of a last entry of the at least one row.

14. (Original) The computer readable medium of claim 8, wherein the generalized scalar function in combination with the column function allow the operation of the column function to be performed for the indeterminate number of entries in the at least one row.

15. (Previously Presented) A system for utilizing a column function for a relational database in a structure query language (SQL) environment, the relational database utilizing data including

a plurality of entries being organized into at least one column and at least one row, the system comprising:

a column function capable of performing an operation on an indeterminate number of entries;

a generalized scalar function for simulating a column environment for the at least one row using the generalized scalar function to allow the at least one row to be provided to the column function as though the at least one row was a column such that the column function can perform an operation the at least one row to provide at least one output;

an interface for allowing a user to specify the at least one row as an argument for the generalized scalar function.

16. (Original) The system of claim 15, wherein the generalized scalar function further fetches a row of the at least one row and provides the row to the column function as though the row was a column.

17. (Original) The system of claim 15, wherein the column function further performs an operation on each of the at least one row to provide an output.

18. (Original) The system of claim 15, wherein the column function provides a maximum of each of the at least one row.

19. (Original) The system of claim 15, wherein the column function provides a minimum of each of the at least one row.

20. (Original) The system of claim 15, wherein the column function performs the operation by performing an initialization phase in response to a first entry of each of the at least one row, performing an evaluation phase on each entry of the at least one row and performing a finalization phase after evaluation of a last entry of the at least one row.

21. (Previously Presented) The system of claim 15, wherein the generalized scalar function in combination with the column function allow the operation of the column function to be performed for the indeterminate number of entries in the at least one row.

22. (Previously Presented) The method of claim 1, wherein each of the plurality of entries corresponds to an intersection of one of the at least one row and one of the at least one column.

23. (Previously Presented) The computer-readable medium of claim 8, wherein each of the plurality of entries corresponds to an intersection of one of the at least one row and one of the at least one column.

24. (Previously Presented) The system of claim 15, wherein each of the plurality of entries corresponds to an intersection of one of the at least one row and one of the at least one column.

25. (Previously Presented) A method for utilizing a column function for a relational database in a structure query language (SQL) environment, the column function capable of performing an operation on an indeterminate number of entries, the relational database utilizing data including a

plurality of entries being organized into at least one column and at least one row, each of the at least one row including a plurality of entries, the method comprising the steps of:

allowing a user to specify the at least one row as an argument for a generalized scalar function;

simulating a column environment for the at least one row using the generalized scalar function to allow the at least one row to be provided to the column function as though the at least one row was a column, the simulating further including using the generalized scalar function to provide the plurality of entries for each of the at least one row to the column function entry by entry; and

performing the column function on the at least one row to provide at least one output.

26. (Previously Presented) The method of claim 25, wherein each of the plurality of entries corresponds to an intersection of one of the at least one row and one of the at least one column.

EVIDENCE APPENDIX

None

RELATED PROCEEDINGS APPENDIX

None